

IC technology and ASIC design for the Cray J90 supercomputer

by D. J. Poli
M. S. Berry
J. N. Kruchowski

The decision to use ASIC technology over a full- or semi-custom approach in the design of a computer system is influenced by many factors, and has a significant impact on the design methodology as well as on the completion schedule of the product. The Cray Research J90™ line of 100-MHz supercomputer systems is an example of a system whose performance, cost, and schedule needs drove the designers to an ASIC solution. The J90 comprises varying numbers of ten unique ASICs, each designed in a 0.5- μ m CMOS technology. The largest of the ASICs contains more than 500000 equivalent two-way NAND CMOS gates. The design cycle, including integrated circuit and first-level packaging technology selection, took just over two years from concept to production. This paper presents a brief history of the Cray ELS (Entry-Level Systems) division and discusses some of the decision processes and trade-offs made during the design of the J90 system, including the decision to use ASIC technology, and its effect on the overall design methodology and CAD flow. The design methodology, which utilized a ground-rule-based HDL/synthesis approach, and the physical design of the

chips, which made use of industry-standard and vendor-proprietary tools, are discussed. Finally, conclusions as to the applicability and the success of utilizing an "off-the-shelf" ASIC technology are drawn.

Introduction

Cray Research has continually been a leader in providing the world's most powerful supercomputing solutions. As the performance of CMOS circuit technology and packaging advanced, Cray Research saw an opportunity to deliver supercomputing solutions to a wider range of customers by offering a supercomputer with a substantially lower price than their high-end machines. A division was established to develop entry-level supercomputers that are binary-compatible with the most powerful Cray Research parallel vector supercomputers [1] and are used to run scientific and engineering applications.

In the fourth quarter of 1992, the entry-level development group at Cray Research was given the task of designing their next-generation supercomputer to replace the YMP-EL™. The YMP-EL is a 33-MHz, air-cooled, CMOS ASIC-based supercomputer with up to four processors and one gigabyte of DRAM-based main memory. The YMP-EL began shipping to customers in March 1992. In July 1993, it was enhanced to allow up to eight processors and four gigabytes of memory. The initial

©Copyright 1996 by International Business Machines Corporation. Copying in printed form for private use is permitted without payment of royalty provided that (1) each reproduction is done without alteration and (2) the *Journal* reference and IBM copyright notice are included on the first page. The title and abstract, but no other portions, of this paper may be copied or distributed royalty free without further permission by computer-based and other information-service systems. Permission to *republish* any other portion of this paper must be obtained from the Editor.

0018-8646/96/\$5.00 © 1996 IBM

Table 1 Table of J90 ASIC gate count and static timing results at worst-case process, 3.135 V, 60°C.

<i>ASIC name</i>	<i>ASIC function</i>	<i>Gate count</i>	<i>Synopsys timing (ns)</i>	<i>IBM final timing (ns)</i>
MBI	DRAM memory interface	51,700	8.0	7.7
MAD	Memory side of memory crossbar for read data	455,400	9.5	9.1
MAR	Memory side of memory crossbar for write data	546,600	11.1	9.2
VA	CPU side of memory crossbar for write data	464,900	10.4	11.0
VB	CPU side of memory crossbar for read data	547,100	10.1	9.9
CI	Channel interface (I/O)	301,900	10.24	10.4
JS	Shared registers for multi-CPU applications	489,500	10.2	9.7
PC	Scalar processor and processor control	695,200	10.4	10.0
VU	Vector processor	562,200	10.6	9.9
MC	Maintenance and clock distribution	24,500	<<10	<<10

goal of the follow-on product, which was ultimately named the J90™, was to design a machine with approximately twelve times the performance of the YMP-EL and with roughly the same entry price of around \$225 000. This performance increase would be achieved by increasing the clock rate to 100 MHz while increasing the maximum number of processors to 32. The J90 [2] would have to be available for customer shipment in the first quarter of 1995.

The project requirements for the J90 drove the development group through a set of decisions regarding integrated circuit technology, design methodology, and first-level packaging that would place them on a path to achieving the cost, performance, and schedule goals. This paper describes some of the trade-offs that were made in the J90 product development and the path that was used to deliver a computer based on 0.5- μ m, 500K-gate, 100-MHz ASICs to market in early 1995.

Custom or ASIC design style

The first decision that had to be made was whether we would stay with an ASIC-based processor design, or attempt to gain a density and performance advantage by using a more custom design of the integrated circuits. Some additional detail on the J90 is necessary to clarify some of the factors that influenced this decision. The central processor is contained on two chips, a scalar processor and a vector processor. The scalar and vector functions had been partitioned onto two chips to meet the gate-count limitations of most 0.5- μ m CMOS technologies (each chip contained 400K to 600K equivalent two-way NAND CMOS gates) and to keep power dissipation under the cooling limit of 25 W. In addition to the two processor chips, eight additional chip types were needed for the system. Four of these were used to create a crossbar network to the large, shared main memory. The remaining four chips were used for I/O control, controlling shared resources, DRAM memory interfacing, and maintenance and clock fan-out. The 100-MHz clock rate was required

by both the core and I/O on all chips in the system.

Table 1 shows the gate counts and the static timing for all ten of the J90 ASICs (estimated timing from the Synopsys Design Compiler™, as well as final post-route timing).

It became obvious that there would be no way to apply a custom design style to ten chips and still meet the schedule. Also, it was concluded that applying a custom design style to the processor chips would not be wise for a number of reasons. First, the entry-level development group was not staffed with enough workers to design custom functional units, or to do chip-level circuit placement and routing in the fifteen months that had been scheduled from the start of the program until the chip set was to begin prototype wafer fabrication. Achieving a higher clock frequency on the processor was not considered an adequate justification for slipping the schedule by even a few months.

An additional consideration when deciding between custom and ASIC processor design is that there is an ongoing CMOS technology curve that continues to provide more performance over time. If a more customized solution caused us to bring our product to market later in time, that product might be competing with ASIC-based machines that were able to capture a later (faster) point on the technology curve.

An additional argument that is often used for doing custom design is that parasitic wire delays can be reduced by squeezing the circuits into a smaller area and hand-optimizing the placement for minimum wire lengths. This advantage appeared to be decreasing. As the metal pitch in CMOS became tighter and as additional metal layers were being added, the average wire length between circuits was decreasing. To determine the effect of wire loading on a critical path within a functional unit, we took the average wire load of approximately 0.25 mm per load (determined by a Rent's rule calculation) and ran static timing on a block of logic. Then we changed the estimated wire load to zero and reran the static timing. Eliminating wire loads completely reduced the worst-case timing by

20%. Given that a customized functional unit design still has some delay due to wiring, we concluded that less than 20% of performance is lost by neglecting customized placement. This was seen as an acceptable trade-off against the time that we would save by using a standard cell, or gate-array ASIC library. It should be noted that this analysis assumes that the physical design of the ASIC is done in a floorplanned fashion in which all circuits within a functional block are placed in the same region on the die.

Using a more custom design style can also result in a reduction in the size of the die that is used. This is a big factor for designs to be used in high-volume production, but it was not a major consideration for the J90 design team, since most part types would see lifetime volumes of under 20000 units.

In summary, time to market was the highest priority of the J90 development project. The decision was made to focus on achieving an optimized ASIC-based design. This eliminated the additional effort and risk of a custom design and removed the need to synchronize interconnection to a processor that is running faster than the rest of the system. An additional priority was to minimize the development resources that the project would require at Cray Research (the entry-level development group consisted of fewer than 40 people). By eliminating the need for circuit library and physical design CAD development, we were able to focus our resources on the design entry, synthesis, simulation, and interconnect design.

Technology supplier selection for silicon and first-level packaging

Selection of the integrated circuit supplier is a critical decision in the development of a high-performance product. The cost, performance, and schedule for the product would all be affected by this decision. It would also be necessary to select a supplier that had an appropriate customer interface model so that we could continue with our existing design methodology. Given the importance of this selection, the J90 development team generated a set of objective metrics that would be used to help determine the optimum technology supplier. Once the set of metrics was defined, the potential technology suppliers were surveyed on how they stood regarding each of the metrics. Table 2 shows some of the metrics used in the technology selection process.

The business items included cost, customer support, services, and the schedule for the proposed technology. In terms of cost, both the NRE (nonrecurring engineering) and unit pricing had a real limit that we were required to meet. The NRE costs had to be met with the budget that had been allocated for the J90 development. The piece part pricing had to be consistent with the requirement that

Table 2 Technology evaluation metrics.

Category	Item	Requirement
Business	NRE	<\$
	Cost per ASIC	<\$
	Eng. sample products	9-1-93
	Production qty.	12-1-93
	Regional design center	yes
	Prototype lead time (weeks)	<10
	Eng. change TAT (days)	<21
Technology	L_{eff} (μ m)	0.5
	Max. usable gate count	>400K
	Die size (mm)	15
	Number of signal I/O	>424
	Power at 100 MHz (W)	<25
	Supply voltage (V)	3.3
	On-chip clock skew (ps)	<300
	SPICE models for Si	yes
	Benchmark frequency (MHz)	>80
CAD	Silicon risk	none
	Place and route	yes
	Qualified circuit libraries	6-1-93
Package	TAB pkg. format	yes

the product would have a market entry price of \$225 000. For customer support, we were looking for a supplier with a design center located in the midwest. In our previous experience with ASIC design, it had been very important to have engineering support that was within a couple hours of travel time. An additional service that has been critical in reducing time to market is rapid turnaround time (TAT) on engineering change orders (ECOs). The final metric in the business category required that the proposed technology be one that would have engineering samples in September of 1993 and production quantities available by December of 1993. This would ensure that the technology was mature by the time that we required production quantities in late 1994.

Evaluating the technology of an ASIC product boiled down to meeting the required gate density, speed, power dissipation, and I/O count. The J90 had already been partitioned to include ASICs with more than 400K gates. This required that we select a technology that provided this number of gates on a die size that would meet the cost requirements. Given the high gate count per chip, it was necessary to select a 3.3-V ASIC technology to keep the power dissipation at a level that could be air-cooled. Ruling out 5-V technologies narrowed the field of potential suppliers a great deal. To evaluate the speed of a potential ASIC solution, gate-delay and wire-loading information was obtained for each of the proposed technologies. This information was used to time three benchmark circuits that had been chosen. The benchmark paths consisted of a path of four-way NAND gates in

series, a path of 4:1 muxes in series, and a path of full adders in series. Each path began and ended with a register, and each gate had a fan-out of four. Although it was a very minimal analysis of the performance of the ASIC technology, the timing data from these three paths gave us a good indicator of the performance of both the technology and the circuit library. For example, the results showed that the IBM technology benefited a great deal by having four power levels for each circuit type. The four power levels allowed better optimization between the gate drive strength and the load that was being driven. This allowed the IBM technology to achieve better performance on the benchmark paths than technologies that had faster raw gate delays but provided only one or two drive strengths for matching the circuit to the load.

In addition to selecting an ASIC supplier that would meet the business and performance objectives of the J90 project, we would need the supplier to match up with our existing CAD methodology. Fortunately, we had a fairly standard CAD design flow. Verilog[®] was used for design entry and simulation, and the Synopsys Design Compiler was used for synthesizing to the gate level, for doing static timing analysis, and for scan insertion. We would require that the supplier be able to accept a gate-level netlist in either an EDIF or Synopsys format. The supplier would then be expected to take the gate-level netlist and perform placement and routing on the design and to close on timing.

The final area of consideration was the first-level (IC) packaging. Our analysis of the card-level interconnect had led us to conclude that we would need a surface-mount packaging technology. The card would require very dense signal routing because of the large number of high-pin-count ASICs. Surface-mount ASIC packaging does not require the large card via needed for pin-through-hole packaging, thus preserving valuable signal routing resources inside the card. Since Cray had a great deal of experience with TAB assembly, we initially concluded that we would use that package format for the J90.

After information was received from all of the potential ASIC suppliers, it was compiled into a table for comparison purposes. We did not necessarily expect that one supplier would meet every requirement, but rather that we would determine the closest match. The data in the table showed that the IBM CMOS 5L technology [3] was the best fit for the J90 program. This technology easily provided the gate density that we required, and the performance on our benchmark circuits achieved the desired 100 MHz. The IBM CAD methodology included the Verilog and Synopsys libraries as well as placement, routing, and optimization services. On the business side of the comparison, the cost and schedule for the 5L technology were in line with our requirements; however, there was no regional design center, and the 21-day ECO

service did not appear to be available. We decided that these two items were not critical, given that IBM 5L was the best fit for the program. We could work with the IBM Burlington design center, and request that IBM provide some form of reduced-turnaround-time reruns for making final logic fixes prior to ramping into production.

After selecting IBM as our supplier, we decided to revisit our choice of first-level packaging. IBM was the only supplier we found that had a surface-mount package with more than 440 signal I/Os. After looking at the 32-mm (625 balls) ceramic ball grid array (CBGA) package that was planned as a standard 5L package offering, we concluded that we would make a switch to that solution despite the fact that significant work had already been done at Cray for use of a TAB package. The CBGA provided a number of compelling benefits, including improved electrical characteristics with an impedance-controlled path between the ball and the I/O cell and a low-inductance power and ground distribution. The CBGA also allowed us to reduce cost by selecting the die size for each ASIC option according to the number of gates in the design, as opposed to the design of the TAB inner lead bond. We went from assuming that we would use a 15-mm die size for eight of the ten options with a TAB package to using die sizes ranging from 9 to 12.7 mm in the CBGA package. The CBGA also provided for a faster card assembly process and allowed a part to be removed from a card, have new solder balls applied by IBM, and then be reused. The TAB design required scrapping any parts that had to be removed from a card. By taking a first-level packaging reset at Cray, we were able to avoid any die design, assembly, and test development at IBM for our unique TAB package, and were able to achieve an improved cost and performance package solution for the J90 [4].

ASIC design constraints

The design of the J90 ASICs began with the development of a rigid set of ASIC design constraints which were internally referred to as design "ground rules." As with all development efforts, requirements that are well defined at the beginning lead to a better design in the end. The design team arrived at a set of ground rules which incorporated circuit requirements, intraboard and interboard requirements, and overall system requirements, many of which were to be enforced at the ASIC level. The general philosophy of the ASIC ground rules was that the synthesis would be done from the perspective that a large (500K-gate) chip was really a collection of independent functional blocks (subchips) interconnected on a piece of silicon. A floorplan of one of the ASICs, which demonstrates the idea of a "system on a chip," is shown in **Figure 1**. Synthesis ground rules would be established for optimizing within a block as well as connecting between

blocks, or to I/O cells. The goal was to achieve accurate pre-physical design timing and optimization, since the schedule did not allow for iterating between design and layout. Some of the ground rules are discussed here, or in more detail as we describe the methodology.

We considered the problem of the length of the "global" wires. Global wires are defined as wires that connect the hierarchy blocks at the highest level. Our specification was that global wires would be long, perhaps from one corner of the die to the other. To account for this assumption, the wires had to meet the following requirements: The macro driving the wire must be high-powered; the pre-physical design timing on any path that included an interblock wire must have a 2-ns guard band to account for the additional delay of the long wire; no logic could exist between the driving macro and the receiving macro; and interblock wires must be single-source, single-sink nets. These restrictions left little opportunity for the logic designers to introduce a timing problem due to interblock wires.

Controlling the maximum load on a net was attempted during synthesis by limiting the maximum fan-out a net could have. This did not necessarily correct all problems, since physical design (PD) would often introduce significantly more wire capacitance than had been anticipated, thereby causing either maximum-capacitance violations or slew violations. Both types of errors were reported by the static timing tool, and were ultimately corrected by repowering a circuit to drive the larger load, or, in some cases, by modifying fan-out trees to accommodate a larger load.

The effect of clock skew was not really seen until after physical design, so a constant "worst-case" value was inserted into every latch-to-latch path during synthesis. The reason for the uncertainty was that since we were doing a true ASIC design, there was no predefined clock tree, nor was there the time or the human resource available to do manual clock balancing; the total clock skew was to be determined by the accuracy of the clock-optimization tool in balancing loads at each level of the tree (pre-PD), and by the ability of the placer and the router to achieve roughly equal-length routes for all macros at a given level of the tree. The clock skew number used as an adder in synthesis was the same number given to the PD team as a worst-case post-PD target between all latches on the chip. This produced slightly conservative synthesis timing, since the probability of a single logic path containing the two latches on the chip with the maximum skew between them is very low.

On the basis of system (ASIC-to-ASIC) timing requirements, as well as the capabilities of the ASIC library we were using, we determined that the timing allowance for the ASIC I/O would be a maximum of 1 ns to enter the ASIC (including receiver delay and latch

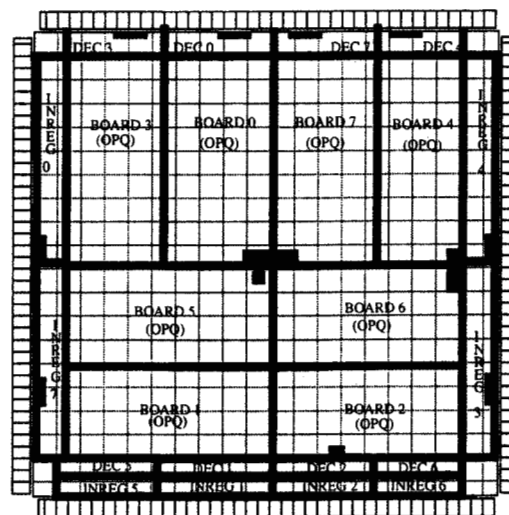


Figure 1

Floorplan of a typical J90 ASIC.

setup time) and 3.8 ns to leave the ASIC (including latch delay, boundary scan mux delay, and driver delay). These assertions were given to the static timing tool, so that we could monitor the I/O timing at all phases of the physical design process (pre-PD, with a flat wire-load model that represented the average wire length for a design done without a floorplan; post-place/pre-route, with wire length estimates based on Manhattan distance between placed macros; and post-route, with actual wire lengths extracted from the layout). Adherence to these numbers required some careful macro placement during the initial phases of physical design.

The signal interconnect between ASICs had to meet the same 10-ns latch-to-latch timing as the internal paths. The 100-MHz interconnection across PC board traces that were more than 20 inches long required that the ASIC output drivers have a controlled output impedance that we could match to the PC board line impedance providing a source termination. The ASIC output driver circuits should also have a small variation in drive current (di/dt) from best-to worst-case process, so that there is adequate drive at worst-case process without excessive switching noise at best-case process. The IBM 5L output circuits met these requirements, since they had been designed with a process compensation circuit that minimized the variation of output impedance and di/dt across the process window. To ensure that we met the 10-ns constraint on all paths between ASICs, we developed a method for system-level

static timing. This involved using internally developed scripts to combine the ASIC I/O time from the post-route static timing reports with the simulated PC board signal trace delay to generate a report which contained the latch-to-latch timing for all paths in the system.

Coding

Our designs were written in Verilog HDL. A set of coding guidelines was produced that instructed the designers on correct coding styles and warned of pitfalls. Prior to this development project, Cray had used a Boolean coding methodology to capture designs. Essentially, this was a gate-level coding technique. The introduction of an outside vendor's tool for design capture was a bold step for Cray, especially because the design was captured at a higher level than had previously been done at Cray, and a synthesis tool (in our case, the Synopsys Design Compiler) was used for gate generation. As a point of interest, we did develop a Boolean-to-Verilog translator to capture some of the previous design efforts that were done in Boolean.

We found it extremely important to keep the coding style very basic. It has been our experience on this project, as well as past projects, that in order to obtain predictable results, code must be kept simple. In general, by using a high-level description of logic to perform a basic synthesis, one should be able to achieve 90% of budget (either timing or area). This implies that there is some knowledge of the structure that will result after synthesis, and therefore means that the HDL must be written in such a way as to guarantee that the synthesis tool will produce the desired results. Care must be taken, for example, to avoid writing HDL that causes the synthesis tool to infer latches that should not exist.

In many cases, synthesis would not lend itself well to doing a custom implementation of logic, since the exact structure of the logic is basically unknown until after synthesis; even then, the structure is not always what may have been anticipated. For example, a regular structure, such as an adder, may be very easy to envision and to lay out, since there is a high degree of parallelism among all the bits of the adder. The same adder, when synthesized, may not have this level of parallelism. In fact, each bit could be constructed differently. This places some limits on the ability to do custom layout of synthesized logic. If custom layout is a requirement, it is possible that synthesis should not be used; or if it is, it would have to be much more carefully controlled than would be necessary for an ASIC implementation.

We found that there were occasions when we needed to instantiate gates in order to guarantee a specific structure after synthesis. This was specifically true in fan-out trees. In addition, the technology we used had a required clocking scheme for an array of registers that was different

from the rest of the system. To control this, we created a function (or predefined gate assignment) that instantiated the required structure. These are two examples of instances where synthesis is not possible, or at least is not the best solution to a problem.

Hierarchy

Hierarchy is a natural method of coding when dealing with large ASICs. The ability to instantiate multiple occurrences of a single design and build multiple levels of hierarchy goes hand-in-hand with the designer's thought process. In our designs we attempted to keep the number of levels of hierarchy between three and four, if for no other reason than to keep names to a reasonable length.

One of the original goals of designing hierarchically was to do formal floorplanning of all of the designs, so that the IBM design center would have a starting point for the placement of the design. Also, since the ground rules were based on a hierarchical design, it seemed necessary to try to follow the hierarchy during placement as closely as possible in order to have the best chance to meet the original timing goals with as little modification to the design as possible. Since the logical and physical hierarchies were identical (with the possible exception of the test logic, which was not timing-critical anyway), there was really no additional effort on the part of the designers to create the physical hierarchy. Because of schedule constraints and the lack of an acceptable tool (for 500K-gate designs in early 1994), formal floorplanning was not actually done on any of the ten ASICs. Some compromise solutions to floorplanning were used during placement with varying degrees of success. This topic is discussed later.

Note that in our method of hierarchy, the test logic is contained in its own level of hierarchy. This was done to buffer the designer from the test insertion process. It also allowed the project to have a single focal point on test issues. Since the test logic was not timing-critical, it did not require the same restrictions as the functional logic.

Verification

Verification occurs at several levels during the design. We have three main functional simulation tools; each served a unique purpose and fulfilled our needs successfully. The designers used Cadence Verilog-XLTM for quick turnaround, initial looks, and single-array simulations. For larger simulations, where we expected multiple tests on a set of designs, Chronologic VCSTM was used. Normally we used this tool when we felt that the additional compile time of VCS justified the savings in memory usage, or when we were incorporating test structures in the simulation. An internally developed simulation tool, Gensim, was used to do full gate simulation of a system configuration. As a point of interest, today we run this tool on a J90.

Chrysalis Design VERIFYer[®] is a formal verification tool that was used to logically compare different versions of an ASIC gate-level netlist. We used this tool to verify that the different versions of the netlist did not change functionally during the design process (i.e., test insertion, place, route, etc.). This allowed us to proceed without having to resimulate at each step.

There are many opinions about which simulator is the best, which is faster, and which is right for the job. Our experience shows that in a large RTL design in which there is a high level of confidence, Chronologic is the best tool, solely on the basis of the fact that after the compile the memory usage is low. The drawback, of course, was a long compile time. It should be noted that since completion of this project, the compile times of the Chronologic tool have been greatly reduced. Verilog proved to be a good all-around simulator for the what-ifs, specific sequence, and quick-turnaround-type simulations. The disadvantage of this approach is the large memory usage throughout the execution of the run. Gensim has its own purpose as a full-system simulation environment in which diagnostics can be run.

Synthesis

Our choice of a synthesis and timing verification tool was Design Compiler from Synopsys. The methodology that we put into place centered around this choice. Since the real success of our project depended on our ability to produce a good design as soon as possible (but not necessarily first-pass correct), it meant that the synthesis tool would have to work or be made to work quickly. This included the ability to support submicron design requirements. There are several elements of our designs that made them somewhat challenging for Synopsys. For instance, our designers wanted to write their HDL in the context of flip-flops (FF), but the 5L technology utilized LSSD-type latches. We resolved this issue by doing two separate mappings, one during test insertion and one in the IBM design center. The fake FF is the model the designers used; it operated on a single clock, with a single data input. The timing model reflected the correct timing and load values for the actual LSSD macro. The model, as inferred by the HDL code, used the "always @(clk)" construct. Functionally it was indeed a flip-flop.

During test insertion, the testable fake FF was inserted to replace the original fake with a testable model. This testable model was the start of the transformation from a flip-flop to an LSSD latch model. The model still contained only one clock, but now it had two clock-enable connections. It also now contained the scan connections required for scan testing. It still consisted of only one macro.

When the design was delivered to the IBM design center, the final transformation was performed: The clock splitter was pulled from the macro, and the two LSSD clocks were reconnected between the splitter and the latches. The end result was two macros, a clock splitter, and an LSSD latch combination (an L1/L2, or master/slave latch). In the final design, there was actually only one clock splitter for every ten or twelve latches. This minimized the wiring and area overhead of utilizing an LSSD design style, while still allowing all of the benefits of LSSD as it pertains to test.

Early in the development cycle, it was decided that an industry-standard, off-the-shelf test solution would be implemented. The choice was made to adhere to the IEEE JTAG 1149.1 standard. Although the Synopsys Test Compiler[™] provided a good JTAG insertion process, including the building of the various elements, we had to have an LSSD-compatible design. The model at this point was not available from Synopsys or from IBM. To work around this problem, we had to design our own controller and boundary scan registers. Using the Synopsys Test Compiler, we were easily able to substitute our models for the Synopsys models to be included automatically during JTAG insertion.

Physical design

Our design goal called for a maximum clock skew of 300 ps between any two latches on a single ASIC. To accomplish this, the clock nets were tuned in several places along the process. Initially, when the tree was installed, it was balanced to achieve an equal number of latches. This gave us a good starting point for the clock tree. Before physical design, the nets were traced and then deleted. The ASIC was then placed, and the tree reconnected and balanced based on the capacitive load on the clock drivers and splitters. Since the tree was deleted before placement, the clock connections were not able to affect the placement of the logic (because the placement was done with a strong emphasis on connectivity). During the rebuilding of the tree after placement, logically equivalent nodes were allowed to be swapped in order to optimize the balancing of the tree and to minimize the routing resources required.

On the basis of our previous experience designing 50K to 100K-gate ASICs, we anticipated that perhaps our greatest challenges for the J90 project would be working with the IBM design center to achieve an optimized placement on 500K-gate designs. As mentioned previously, although advanced floorplanning on the large designs was thought to be important to successfully wiring the design and closing on timing, we would not be able to include formal floorplanning because of schedule constraints and the lack of an acceptable tool. As a compromise to floorplanning, several placement approaches were used by

Table 3 Hardware, software, and staffing requirements.

Hardware

- Cray YMP 8 CPU supercomputer (simulation)
- Sun SPARC 10 for each ASIC option type
- Sun 690M 4 CPU server with 1GB memory
- 12 GB disk storage

Licenses

- 4 Synopsys design experts, 1 test compiler
- 3 Verilog
- 3 Chronologic

People

- 11 logic designers
- 6 CAD/simulation engineers
- 6 circuit designers
- 6 mechanical designers

the IBM design center in an attempt to retain some of the logic hierarchy. One method was to have the placement tool ignore the global nets (between blocks). Since these nets were not included as part of the "cost function" for placement, the placer tended to keep all circuits within a block grouped together. This approach worked well on some, but not all, designs. In some cases there were wirability problems with the placement, since there were no global wires to spread out the circuits. Another approach to placement was to apply capacitance targets for the placement to meet. Small-capacitance targets were applied to circuits that drove local nets, and large-capacitance targets were applied to global nets. Again, this approach provided mixed results. Ultimately, after a great deal of place-and-route CAD development, the design center was able to successfully complete placement and routing, and close on timing for all ten of the ASIC designs. Through the course of the second and third quarters of 1994, the physical design of the ASICs was completed, and they were released to the foundry for prototype fabrication.

Conclusions

Because of the strict requirements imposed on the design from the beginning, the correlation between pre-PD and post-PD timing was excellent; we achieved less than 10% deviation from synthesis timing to final hardware timing. This was accomplished by ground-rule enforcement through the use of the Synopsys Design Compiler, and teamwork among CRI, Synopsys, and the IBM design center.

Since the completion of this project, floorplanning CAD tools have improved a great deal and would be a benefit to successfully designing high-performance ASICs. Lack of floorplanning on the J90 ASICs forced us to use conservative pre-PD static timing assumptions and select die sizes that would yield moderate levels of gate utilization. For example, one of the J90 ASICs was

originally targeted to a die that was 10 mm on a side. After a three months' attempt to complete placement and routing, the design was retargeted to a 12-mm die, where it was still found to be challenging to route. A good floorplanning process would allow gate utilization and routability to be studied in advance of the final PD so that the project schedule is not affected.

The ASIC design methodology utilized in the J90 proved to be the optimum choice for this project. We were able to target a developing silicon technology and leverage the circuit and CAD library development work of the supplier. It also allowed us to focus on logic and interconnect design while delegating the physical design to the foundry. The performance advantage that a custom IC design approach could yield was not a strong enough incentive to offset the additional development cost and time-to-market delay that would be incurred.

Table 3 shows the number of vendor licenses, design engineers, and CAD engineers used to complete the project as well as the amount of disk space and type of hardware used.

After receiving the final prototype option from IBM in December of 1994, the J90 was powered up, and within hours we were able to boot the Cray UNICOS™ operating system; however, ECOs to three of the ASICs would be necessary before the J90 would be production-ready. Two of the logic bugs had been identified in simulation prior to the December power-up; therefore, the ECOs were already in process at IBM. The third ECO required only a handful of edits to the signal wires on the chip, which allowed IBM to simply modify the existing physical design, then begin fabrication of the new design. By requesting that IBM build production quantities from the prototype run of the ASICs that required ECOs, J90 production was able to begin immediately after the new designs were received. In March of 1995 the first J90 was shipped to a customer. With approximately 8, 20, 30, and 40 machines shipped in March, April, May, and June, respectively, the J90 saw the fastest production volume ramp-up of any machine that has been produced by Cray Research. Cray Research shipped approximately 250 J90 computers in 1995.

As a final note, it seemed to be universally felt by the development team that the J90 development project was the most successful program of any that they had experienced. The close working relationship between the individuals on the development team and between the team and its key suppliers resulted in a two-year design cycle for a product that utilized leading-edge technology.

Acknowledgments

The authors would like to express their appreciation to Steve Woods, CAD engineer at Cray Research, for his assistance in describing the coding and verification

methodologies used in developing the J90. The authors would also like to thank Gene Sluss of Fujitsu Microelectronics, formerly of Cray Research, for his systematic approach to evaluating ASIC technology.

Cray, J90, YMP-EL, and UNICOS are trademarks of Cray Research Inc.

Design Compiler and Test Compiler are trademarks of Synopsys, Inc.

Verilog is a registered trademark, and Verilog-XL is a trademark or registered trademark, of Cadence Design Systems, Inc.

VCS is a trademark of Viewlogic Systems, Inc.

Design VERIFYer is a registered trademark of Chrysalis Symbolic Design, Inc.

References

1. M. Gokhale and W. Carlson, "An Introduction to Compilation Issues for Parallel Machines," *J. Supercomputing* **6**, 283-314 (1992).
2. Cray Research web site: <http://www.cray.com/PUBLIC/>.
3. R. Gregor, C. Ng, J. Libous, E. Carter, R. Beaudoin, A. Chu, D. Grindel, J. Kinney, M. Lee, L. Montes, J. Oppold, M. Russell, A. Secor, and G. Yenik, "A One-Million-Circuit CMOS ASIC Logic Family," *Proceedings of the IEEE 1993 Custom Integrated Circuits Conference*, May 1993, p. 23.1.1.
4. D. Carlson, "Cray J90 Computer System Air-Cooled Supercomputer Packaging," *Proceedings of the 1994 International Electronics Packaging Conference*, September 1994.

Received October 27, 1995; accepted for publication April 3, 1996

David J. Poli *Cray Research Inc., P.O. Box 4000, Chippewa Falls, Minnesota 54729 (dpoli@cray.com)*. Mr. Poli received his B.S.E.E. from the University of Minnesota, Minneapolis, in 1985. From 1985 to 1991 he worked for Unisys Corporation in the 2200 series computer systems group, where he initially worked as a signal integrity engineer and later designed a custom ECL clock fan-out and control chip. Since 1991, Mr. Poli has worked for Cray Research Inc. doing ASIC and system-level circuit design on the YMP-EL and J90 series CMOS, parallel vector supercomputers.

Michael S. Berry *Cray Research Inc., P.O. Box 4000, Chippewa Falls, Minnesota 54729 (mberry@cray.com)*. Mr. Berry received his B.A. in physics from Drew University, Madison, New Jersey, in 1989, and his M.S.E.E. from the New Jersey Institute of Technology (NJIT), Newark, in 1993. From 1992 to 1994, he worked for NJIT as a research engineer in the Class 10 Wafer Fabrication Laboratory, where he was responsible for all aspects of CMOS wafer processing. Since 1994, Mr. Berry has worked for Cray Research, Inc., doing ASIC and system-level circuit design, including static timing analysis and design methodology development, for the J90 series of CMOS, parallel vector supercomputers.

James N. Kruchowski *Intel Corporation, 1900 Prairie City Road, Folsom, California 95630 (james_kruchowski@ccm.fm.intel.com)*. Mr. Kruchowski received a B.S.E.E. degree, with distinction, from the University of Minnesota in 1980. From 1980 to 1986 he worked as a plasma etch-photolithography IC process development engineer in the VHSIC development program at Honeywell, and also helped to start up a new IC process fabrication facility with Sperry-Unisys. In 1986 he joined Cray Research Inc. to develop a multichip module process, and shortly thereafter became a system circuit design engineer for the C-90 high-speed computer system. In 1990, Mr. Kruchowski joined the Cray Research Entry-Level Systems design effort as head circuit design engineer, where his team was responsible for IC circuit design, system circuit design, and clock distribution designs for a new series of high-speed, low-price/power computer systems at Cray. He is currently involved with a high-speed interconnect design effort within Intel Corporation.